Quadrilateral Mesh Smoothing

Xuan Huang

February 7, 2018

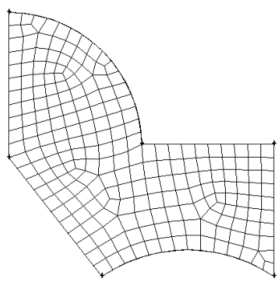

Figure 1: A quadrilateral mesh

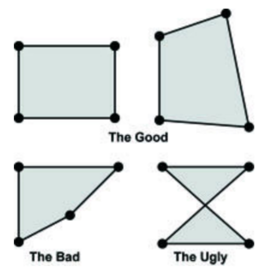

Figure 2: the good, the bad, and the ugly

1 Introduction

Maybe give some examples, like "practical applications as xxx"

Mesh quality improvement is an important problem with a wide range of practical applications. Element quality of a mesh heavily affects the results of numerical simulation done using that mesh. In the context of finite element mesh smoothing, vertex repositioning is the primary technique employed, where we allow tangential vertex motion only and the edges of the mesh is unchanged.

What's a numerical simulation? Where might it be applied generally?

2 Element Quality

Element quality is measured either by max/min angles or aspect ratio (longest edge over shortest), or both. In a quadrilateral mesh the aspect ratio is defined as the largest ratio of longest edge over shortest edge among all elements, and max/min angles are the maximum/minimum among all angles. These two proprieties are related in triangles (by

Nice! Concise and clear definitions!

the law of sine), but the relationship is quite loose in quads (Figure 3). We investigate a smoothing method based on the idea of improving aspect ratio by circumscircles.

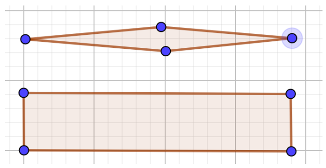

Figure 3: Aspect ratio and angles are not necessarily related

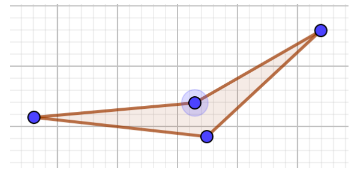

Figure 4: Non-convex quad

Convexity is another important aspect of quadrilateral elements (Figure 4). It is very undesired to generate any element that has an angel greater than 180 degree. Again this is never a problem for triangles, and one of our goals will be trying to eliminate such cases from happening.

3 Background

Existing methods usually divide into mesh modification via vertex insertion/deletion, edge/face swapping and remeshing [Dey and Ray 2010], or vertex repositioning without changing mesh connectivity [Amenta et al. 1997; Field 1988; Zhou and Shimada 2000]. Among those that do not modify mesh topology, Laplacian smoothing is the most commonly used because of its simplicity. In its most basic form, it moves each vertex to the barycenter of its neighboring vertices. It is a local method with very low computational

not sure if center of gravity may be clearer wording

cost, compared to the alternatives, which are optimization-based [Chen and Holst 2011; Freitag 1997; Parthasarathy and Kodiyalam 1991]. e most closely related work is Zhou and Shimadas angle-based Laplacian smoothing [Zhou and Shimada 2000], which is a variant of Laplacian smoothing.

4 Current Plan

Nice! Clear plan as what you will do!

We plan to develop a computationally inexpensive smoothing algorithm for quadrilateral meshes. The goal is to first maintain the convexity, and then to provide a generally effective improvement on arbitrary input mesh. The process will be mostly experimental and the final result will be reflected on the outcome of the algorithm running on various examples. Likely the result from angle-based method will also be presented for comparison.

Is this the previous result? Then what's the main method you are exploring now?

There are three major problems to consider at this point. First of all, We have to decide how to deal with the two triangles once the quad has been split. The original method we applied for triangular meshes moves each vertex on it circumscircle so that the aspect ratio of that single triangle will be improved. And since the circumscirlce also indicates the value of the angles, the other option is to try emerging the two circumscircle in someway, so that they become more similar to each other (if they overlap completely then we can possibly get a square by sliding vertices along the circumscircle). graphs here might be helpful in clarification?

Secondly, we have to consider how to divide the quads into two triangles. Most of the time different divisions will result in different circumscircles, and we have to find a general rule of which way of splitting the quad is more desired.

The order of smoothing also has to be determined. We can always sort the quads by their quality and start with all vertices on the worst one. Or we can go by vertex and determine the most urgent vertex to move at every step. Since we are doing thing locally, the priority has to be measured in some way that will guarantee a better result for the overall mesh.

how do you determine the urgency of a vertex?

5 Documentation

L. Chen and M. Holst. 2011. Efficient Mesh Optimization Schemes based on Optimal Delaunay Triangulations. Computer Methods in Applied Mechanics and Engineering 200 (2011), 967984.

T. Dey and T. Ray. 2010. Polygonal Surface Remeshing with Delaunay Refinement. Engineering with Computers 26 (2010), 298301.

D. Field. 1988. Laplacian Smoothing and Delaunay Triangulations. Communications in Applied Numerical Methods 4 (1988), 709712. Issue 6.

L. Freitag. 1997. On Combining Laplacian and Optimization-based Mesh Smoothing Techniques. AMD Trends in Unstructured Mesh Generation 220 (1997), 3743.

V. Parthasarathy and S. Kodiyalam. 1991. A Constrained Optimization Approach to Finite Elemement Mesh Smoothing. Finite Elements in Analysis and Design 9 (1991), 309320.

T. Zhou and K. Shimada. 2000. An Angle-based Approach to Two-dimensional Mesh Smoothing. In Proceedings, 9th International Meshing Roundtable. 373384.